Classification problems in object-based representation systems

Amedeo Napoli
LORIA – UMR 7503
B.P. 239, 54506 Vandœuvre-ls-Nancy Cedex, France
(Email: {napoli@loria.fr})

Abstract

Position paper for the DL'99 Workshop at IJCAI'99.

1 Introduction

Classification is a process that consists in two dual operations: generating a set of classes and then classifying given objects into the created classes. The class generation may be understood as a learning process and the determination of an appropriate set of classes is an aspect of building a knowledge-based system. Object classification may be understood as a problem-solving process, i.e. recognizing an unknown object by identifying its characteristics in searching for the reference classes to which the object may be related. The classification process is made more efficient when classes are organized into a hierarchy.

A hierarchy of classes is also the basis of object-based representation systems (OBRS) where knowledge is represented within classes and classification is used for reasoning, as in description logics [Nebel,1990] [Donini *et al.*,1996]. The goal of this position paper is to introduce briefly the notions of OBRS, of classification problems in OBRS, and to show how classification problems can be solved in an OBRS using a case-based reasoning paradigm. OBRS present a number of similarities with description logics. Moreover, the results of theoretical and practical studies carried out in the context of description logics can be reused with profit in the context of OBRS.

2 Object-based representation systems

An object-based representation system (OBRS) is based on a *hierarchy* $\mathcal{H}_{\mathtt{kb}} = (\mathcal{C}, \sqsubseteq_{\mathtt{kb}}, \top, \bot)$, which is a directed graph without cycle, where $\mathcal{C}$ is a set of *classes*[1], $\sqsubseteq_{\mathtt{kb}}$ is a partial ordering called *specialization*, $\top$ is the *root* of $\mathcal{H}_{\mathtt{kb}}$ (the maximum of $\mathcal{C}$ for $\sqsubseteq_{\mathtt{kb}}$), $\bot$ is the bottom of $\mathcal{H}_{\mathtt{kb}}$ (the minimum of $\mathcal{C}$ for $\sqsubseteq_{\mathtt{kb}}$). A vertex of $\mathcal{H}_{\mathtt{kb}}$ denotes a class and an edge such as $\mathtt{C} \sqsubseteq_{\mathtt{kb}} \mathtt{D}$ denotes the fact that the class $\mathtt{C}$ is a specialization of the class $\mathtt{D}$ in $\mathcal{H}_{\mathtt{kb}}$.

A class $\mathtt{C}$ represents a real-world concept and is composed of a collection of *attributes* describing the characteristics and the behavior of the concept. *Annotations* may be attached to an attribute to specify the type, the domain, the cardinality and value of the attribute. Formally, a class $\mathtt{C}$ has an identity and is described by a conjunction $\mathtt{C} = (\mathtt{a}_1, \mathtt{s}_1) \sqcap (\mathtt{a}_2, \mathtt{s}_2) \sqcap ... \sqcap (\mathtt{a}_n, \mathtt{s}_n)$, where $\mathtt{a}_\mathtt{i}$ denotes an attribute and $\mathtt{s}_\mathtt{i}$ an annotation. The attributes $\mathtt{a}_\mathtt{i}$ are mutually distinct, but several annotations may be attached to the same attribute and this is denoted by $(\mathtt{ai}, (\mathtt{s}_{\mathtt{i}_1}, \mathtt{s}_{\mathtt{i}_2}, .., \mathtt{s}_{\mathtt{i}_m}))$.

OBRS are used for representing knowledge and for reasoning in order to solve problems in a given domain. The hierarchy $\mathcal{H}_{\mathtt{kb}}$ corresponds to the knowledge base of the system. Reasoning is mainly based on *inheritance* and *classification*. Inheritance controls attribute-annotations sharing in $\mathcal{H}_{\mathtt{kb}}$ and is a basis for default reasoning. Classification makes explicit the dependencies existing between two classes $\mathtt{C}$ and $\mathtt{D}$, or between an instance $\mathtt{x}$ and a class $\mathtt{C}$. The classification process is based on a *subsumption* relation –similar to subsumption in description logics– that is a partial ordering organizing classes in a *subsumption hierarchy* denoted by $\mathcal{H}_{\mathtt{sub}}$. The subsumption relation relies on the following principles: (i) it is defined on a set of primitive and defined classes, (ii) the attributes of a primitive class $\mathtt{C}$ are considered as necessary conditions for an individual to be an instance of $\mathtt{C}$, (iii) the attributes of a defined class $\mathtt{C}$ are considered as necessary and sufficient conditions for an individual to be an instance of $\mathtt{C}$, (iv) annotations attached to attributes may be completed but not overridden.

3 Classification problems in OBRS

The classification process can be used to handle queries in a subsumption hierarchy $\mathcal{H}_{\mathtt{sub}}$ [Borgida and Guinness,1996]. A query is represented by a defined class $\mathtt{Q}$ and is answered by classifying $\mathtt{Q}$ in $\mathcal{H}_{\mathtt{sub}}$. The answer

[1] Also called *frames*.

is composed of the set of the instances of the subsumees of Q in $\mathcal{H}_{\mathtt{sub}}$ plus the instances of the subsumers of Q verifying the constraints given in Q. A query against a hierarchy $\mathcal{H}_{\mathtt{sub}}$ may be seen as an elementary *classification problem*. Various definitions of classification problems are also presented in [Puppe,1993] [Stefik,1995] [Lenz *et al.*,1998].

3.1 The notion of a classification problem

Classification problem-solving consists of first representing a new problem as a defined class and second of classifying this defined class in a subsumption hierarchy. The approach of classification problems presented here is analytical (goal-directed) and the emphasis is on the goal of the problem.

Definition 1 *A* classification problem *consists in searching for a correspondence between elements in an input space, the* data *of the problem, and elements in an output space being classes, the* solutions *of the problem.*

In the framework of OBRS, the problem data correspond to the goal of the problem and are represented within a defined class P. The problem solutions are represented as classes organized in a subsumption hierarchy $\mathcal{H}_{\mathtt{sub}}$. It must be noticed that problem data and problem solutions are represented within the same formalism, namely classes.

A classification problem P can be assimilated to a query and solving P consists in classifying P in a subsumption hierarchy $\mathcal{H}_{\mathtt{sub}}$, the MSS of P in $\mathcal{H}_{\mathtt{sub}}$ providing the elements for solving P. Thus, the content of $\mathcal{H}_{\mathtt{sub}}$ plays an important role on classification problem-solving.

3.2 An organization of classification problems

An attribute-annotation pair $(\mathtt{a}, \mathtt{s}')$ is *stronger* –or *more constrained*– than a pair $(\mathtt{a}, \mathtt{s})$ if and only if $\mathtt{s}' \longrightarrow \mathtt{s}$. For example, $\mathtt{s}' = (\mathtt{x} \geqslant 40) \longrightarrow \mathtt{s} = (\mathtt{x} \geqslant 30)$, or $\mathtt{s}' = [3, 4] \longrightarrow \mathtt{s} = [2, 5]$. The fact that attribute-annotation pairs are comparable allows us to make classification problems comparable:

Definition 2 *The classification problem* P *is* more general *than the classification problem* Q*, or* P *subsumes* Q*, denoted by* $\mathtt{Q} \sqsubseteq_{\mathtt{sub}} \mathtt{P}$*, if and only if for every attribute-annotation pair* $(\mathtt{a}, \mathtt{s})$ *in the defined class representing* P*, there exists an attribute-annotation pair* $(\mathtt{a}, \mathtt{s}')$ *in the defined class representing* Q *such that* $\mathtt{s}' \longrightarrow \mathtt{s}$.

Formally, $\mathtt{P} = (\mathtt{a}_1, \mathtt{s}_1) \sqcap (\mathtt{a}_2, \mathtt{s}_2) \sqcap ... \sqcap (\mathtt{a}_n, \mathtt{s}_n)$ is more general than $\mathtt{Q} = (\mathtt{b}_1, \mathtt{s}'_1) \sqcap (\mathtt{b}_2, \mathtt{s}'_2) \sqcap ... \sqcap (\mathtt{b}_m, \mathtt{s}'_m)$ if and only if for every pair $(\mathtt{a}_i, \mathtt{s}_i)$ in P there exists a pair $(\mathtt{b}_j, \mathtt{s}'_j)$ in Q such that $\mathtt{a}_i = \mathtt{b}_j$ and $\mathtt{s}'_j \longrightarrow \mathtt{s}_i$.

The definition 2 introduce the subsumption relation $\sqsubseteq_{\mathtt{sub}}$ that organize classification problems into the hierarchy $\mathcal{H}_{\mathtt{sub}}$. Moreover, a solution or a method for building a solution may be attached with every defined class in $\mathcal{H}_{\mathtt{sub}}$. Relying on this hypothesis, solving a classification problem Q consists in searching for a problem P subsuming Q in $\mathcal{H}_{\mathtt{sub}}$, and then *adapting* the solution of P to Q, leading from classification-based reasoning to CBR [Melis *et al.*,1998] [Lieber and Napoli,1998]. The subsumption relation between a reference problem P and a new problem Q measures the *similarity* between P and Q.

3.3 From classification to hierarchical case-based reasoning

The goal of CBR is to associate a solution `Sol(target)` with a *target* problem `target` by reusing the solution `Sol(source)` of a known *source* problem `source`–a *past case*– memorized in a *case base*. A case is a pair problem-solution (P, Sol(P)) and a case base is a finite set of cases, $\mathtt{Case\text{-}Base} = \{(\mathtt{P_i}, \mathtt{Sol}(\mathtt{P_i}))/\mathtt{i} = 1, .., \mathtt{q}\}$. CBR relies on a three-step cycle (1) retrieval (of a problem `source` in a case base *similar* to the problem `target`), (2) adaptation (of the solution `Sol(source)` in order to solve the problem `target`), (3) memorization (of the problem `target` and the building characteristics of the solution `Sol(target)`).

The case base can be flat or partially ordered, e.g. by means of a hierarchy of *indexes*. Then, an *indexing process* –automatic or manual– based on an abstraction process associates an *index* idx(K) with a case K = (P, Sol(P)). The index of K is a "summary" of P and encodes the main characteristics of P, i.e. the elements playing a role in the resolution process of the problem P. An index is an abstraction of a case and thus is a generalization of that case, more precisely a generalization of the problem statement associated with that case.

Given a new classification problem `target`, a case (`source`, `Sol(source)`) where `source` is similar to `target`, and adaptable, has to be retrieved in `Case-Base`. In the following, we introduce the notion of hierarchical CBR problem-solving, that can be seen as a classification process combined with an adaptation process (the learning step is not taken into account here):

- *Classification*: the classification problem `target` is classified in $\mathcal{H}_{\mathtt{sub}}$. Every index class in $\mathcal{H}_{\mathtt{sub}}$ subsuming `target`, i.e. $\mathtt{target} \sqsubseteq_{\mathtt{sub}} \mathtt{idx(source)}$, is associated with a case (`source`, `Sol(source)`) whose solution can be reused to build `Sol(target)`, i.e. `Sol(source)` can be adapted to build `Sol(target)`.
- *Adaptation*: depending on the result of the classification process, a *similarity path* between `source` and `target` can be explicited and a solution of `target` can be built accordingly. A similarity path

is defined by a sequence of generalization, specialization, and transformation operations, leading from `source` to `target` in $\mathcal{H}_{\mathtt{sub}}$.

4 Conclusion

In this paper, we introduce the notion of classification problem in an OBRS, and we show how the classification process can be embedded into a CBR problem-solving process. The basic structure allowing classification to support case retrieval is an index hierarchy, considered as a subsumption hierarchy, allowing to take advantage of the classification process for case access and exploitation. Moreover, domain knowledge can be associated with cases in order to improve the CBR problem-solving process, e.g. for adapting past solutions.

References

[Borgida and Guinness, 1996] A. Borgida and D.L. Mc Guinness. Asking Queries about Frames. In *Proceedings of KR'96*, pages 340–349, 1996.

[Donini *et al.*, 1996] F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 191–236. CSLI Publications, Stanford, CA, 1996.

[Lenz *et al.*, 1998] M. Lenz, E. Auriol, and M. Manago. Diagnosis and Decision Support. In *Case-Based Reasoning Technology – From Foundations to Applications*, LNAI 1400, pages 51–90. Springer, Berlin, 1998.

[Lieber and Napoli, 1998] J. Lieber and A. Napoli. Correct and Complete Retrieval for Case-Based Problem-Solving. In *Proceedings of ECAI'98*, pages 68–72, 1998.

[Melis *et al.*, 1998] E. Melis, J. Lieber, and A. Napoli. Reformulation in Case-Based Reasoning. In *Advances in Case-Based Reasoning (EWCBR'98)*, LNAI 1488, pages 172–183. Springer, Berlin, 1998.

[Nebel, 1990] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems.* LNAI 422. Springer, Berlin, 1990.

[Puppe, 1993] F. Puppe. *Systematic Introduction to Expert Systems.* Springer, Berlin, 1993.

[Stefik, 1995] M. Stefik. *Introduction to Knowledge Systems.* Morgan Kaufmann Publishers, San Francisco, CA, 1995.